WHAT IS CLAIMED IS:

1. A method for designing a computer program, comprising:

accessing a plurality of domain rules, each domain rule being invariant;

displaying a plurality of business rules, each business rule being variable;

selecting one or more business rules of the plurality of business rules in response to a user selection;

customizing the one or more business rules;

associating the one or more business rules with a procedure;

associating the domain rules with the procedure;

displaying a model representing the procedure; and

generating a code corresponding to the procedure in order to design a computer program.

2. The method of Claim 1, further comprising:

collecting the domain rules and the business rules;

allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases; and

assessing the domain rules and the business rules in accordance with the realization.

3. The method of Claim 1, further comprising:

checking a syntax of the code; and

providing a notification if a syntax error is detected.

4. The method of Claim 1, further comprising:

checking a logical consistency of the code; and

providing a notification if a logical inconsistency is detected.

5. The method of Claim 1, further comprising:

checking a compatibility between the model and the code; and

providing a notification if an inconsistency is detected.

6. The method of Claim 1, wherein the model is expressed according to a common modeling language.

7. Logic for designing a computer program, the logic embodied in a medium and operable to:

access a plurality of domain rules, each domain rule being invariant;

display a plurality of business rules, each business rule being variable;

select one or more business rules of the plurality of business rules in response to a user selection;

customize the one or more business rules;

associate the one or more business rules with a procedure;

associate the domain rules with the procedure;

display a model representing the procedure; and

generate a code corresponding to the procedure in order to design a computer program.

8. The logic of Claim 7, further operable to:

collect the domain rules and the business rules;

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

9. The logic of Claim 7, further operable to:

check a syntax of the code; and

provide a notification if a syntax error is detected.

10. The logic of Claim 7, further operable to:

check a logical consistency of the code; and

provide a notification if a logical inconsistency is detected.

11. The logic of Claim 7, further operable to:

check a compatibility between the model and the code; and

provide a notification if an inconsistency is detected.

12. The logic of Claim 7, wherein the model is expressed according to a common modeling language.

13. A system for designing a computer program, comprising:

a database operable to store a plurality of domain rules, each domain rule being invariant; and

a server coupled to the database and operable to:

display a plurality of business rules, each business rule being variable;

select one or more business rules of the plurality of business rules in response to a user selection;

customize the one or more business rules;

associate the one or more business rules with a procedure;

associate the domain rules with the procedure;

display a model representing the procedure; and

generate a code corresponding to the procedure in order to design a computer program.

14. The system of Claim 13, the server further operable to:

collect the domain rules and the business rules;

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

15. The system of Claim 13, the server further operable to:

check a syntax of the code; and

provide a notification if a syntax error is detected.

16. The system of Claim 13, the server further operable to:

check a logical consistency of the code; and

provide a notification if a logical inconsistency is detected.

17. The system of Claim 13, the server further operable to:

check a compatibility between the model and the code; and

provide a notification if an inconsistency is detected.

18. The system of Claim 13, wherein the model is expressed according to a common modeling language.

19. A system for designing a computer program, comprising:

means for accessing a plurality of domain rules, each domain rule being invariant;

means for displaying a plurality of business rules, each business rule being variable;

means for selecting one or more business rules of the plurality of business rules in response to a user selection;

means for customizing the one or more business rules;

means for associating the one or more business rules with a procedure;

means for associating the domain rules with the procedure;

means for displaying a model representing the procedure; and

means for generating a code corresponding to the procedure in order to design a computer program.

20. A method for designing a computer program, comprising:

collecting a plurality of domain rules, allocating the domain rules to a plurality of use cases, realizing the use cases, assessing the domain rules in accordance with the realization, and accessing the domain rules, each domain rule being invariant;

displaying a plurality of business rules, each business rule being variable;

selecting one or more business rules of the plurality of business rules in response to a user selection;

customizing the one or more business rules;

associating the one or more business rules with a procedure;

associating the domain rules with the procedure;

displaying a model representing the procedure, the model expressed according to a common modeling language;

generating a code corresponding to the procedure in order to design a computer program;

checking a syntax of the code, and providing a notification if a syntax error is detected;

checking a logical consistency of the code, and providing a notification if a logical inconsistency is detected; and

checking a compatibility between the model and the code, and providing a notification if an inconsistency is detected.

21. A method for managing rules for designing a computer program, comprising:

accessing a plurality of rules;

analyzing the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

storing the business rules; and

providing a business rule from the stored business rules in response to a request for the business rule.

22. The method of Claim 21, further comprising:

customizing the provided business rule;

associating the customized business rule with a procedure; and

generating a code corresponding to the procedure in order to design a computer program.

23. The method of Claim 21, further comprising:

associating the domain rules with a procedure; and

generating a code corresponding to the procedure in order to design a computer program.

24. The method of Claim 21, further comprising:

allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases; and

assessing the domain rules and the business rules in accordance with the realization.

25. A system for managing rules for designing a computer program, comprising:

a database operable to store a plurality of rules; and

a server coupled to the database and operable to:

analyze the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

store the business rules; and

provide a business rule from the stored business rules in response to a request for the business rule.

26. The system of Claim 25, wherein the server is further operable to:

customize the provided business rule;

associate the customized business rule with a procedure; and

generate a code corresponding to the procedure in order to design a computer program.

27. The system of Claim 25, wherein the server is further operable to:

associate the domain rules with a procedure; and

generate a code corresponding to the procedure in order to design a computer program.

28. The system of Claim 25, wherein the server is further operable to:

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

29. Logic for managing rules for designing a computer program, the logic embodied in a medium and operable to:

access a plurality of rules;

analyze the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

store the business rules; and

provide a business rule from the stored business rules in response to a request for the business rule.

30. The logic of Claim 29, further operable to:

customize the provided business rule;

associate the customized business rule with a procedure; and

generate a code corresponding to the procedure in order to design a computer program.

31. The logic of Claim 29, further operable to:

associate the domain rules with a procedure; and

generate a code corresponding to the procedure in order to design a computer program.

32. The logic of Claim 29, further operable to:

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

33. A system for managing rules for designing a computer program, comprising:

means for accessing a plurality of rules;

means for analyzing the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

means for storing the business rules; and

means for providing a business rule from the stored business rules in response to a request for the business rule.

34. A method for managing rules for designing a computer program, comprising:

accessing a plurality of rules;

analyzing the rules to separate a plurality of domain rules from a plurality of business rules, each domain rule being invariant, each business rule being variable;

allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases;

assessing the domain rules and the business rules in accordance with the realization;

storing the business rules;

providing a business rule from the stored business rules in response to a request for the business rule;

customizing the provided business rule;

associating the customized business rule with a procedure;

associating the domain rules with the procedure; and

generating a code corresponding to the procedure in order to design a computer program.